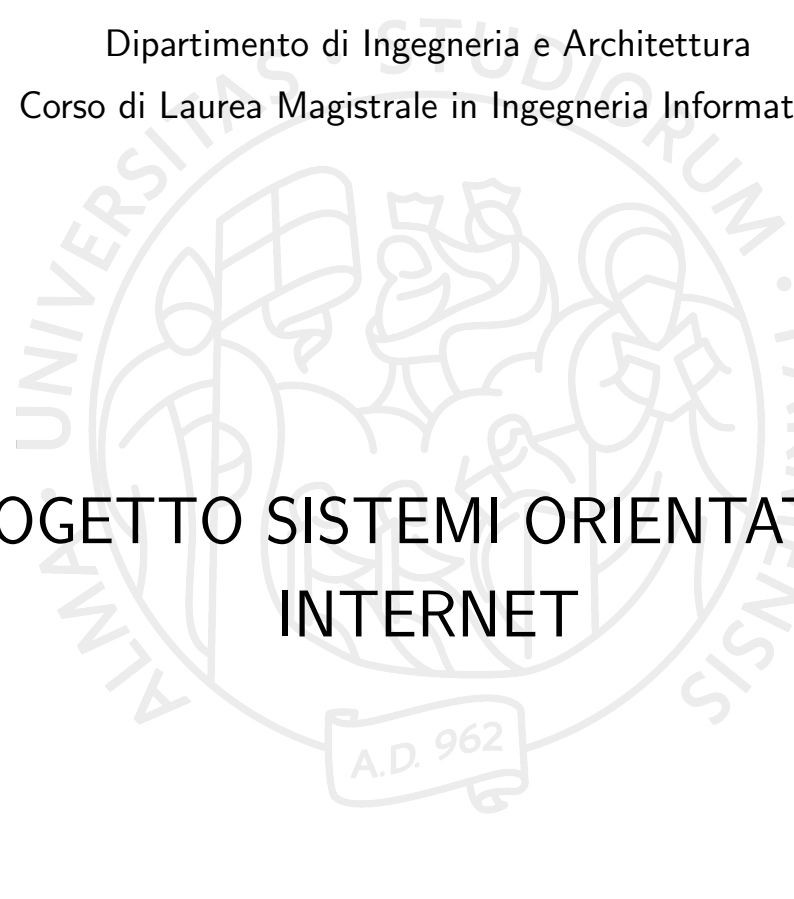


UNIVERSITÀ DI PARMA

Dipartimento di Ingegneria e Architettura
Corso di Laurea Magistrale in Ingegneria Informatica



PROGETTO SISTEMI ORIENTATI AD INTERNET

Professori:

Chiar.mi Prof. DANIELE PORTA, FABIO STROZZI

Studente:

ALEX TIVOLI

ANNO ACCADEMICO 2022-2023

Indice

Introduzione	1
1 Frontend	2
2 Backend	3
3 Microservizi	4
3.1 Attuatore	4
3.2 Pompa di Calore	4
3.3 Porta	5
3.4 Finestre	5
3.5 Tempo atmosferico	5
3.6 Termometro	5

Introduzione

La presente relazione mostra l'architettura dell'applicazione Web sviluppata per il progetto di Sistemi Orientati ad Internet. L'obiettivo del progetto consiste nello sviluppo di un'applicazione Web che simula la temperatura esterna e offre la possibilità di controllare alcune sensore di una stanza di una casa: una porta, una pompa di calore, N finestre.

Sono presenti un server back-end, un servizio per simulare la temperatura del tempo, un servizio per gestire lo stato di una porta, un servizio per gestire lo stato e la temperatura di una pompa di calore, un servizio per gestire lo stato di una o più finestre, un servizio per controllare un termometro posizionato all'interno della stanza, un attuatore che coordina i comandi che un utente esegue ed infine un front-end con cui un utente può interagire.

L'applicazione è disponibile via Http all'indirizzo:

[http://alex-tivoli.soi2223.unipr.it:8080/](http://alex-tivoli soi2223.unipr.it:8080/)

Il codice sorgente dell'applicazione è disponibile alla repository github:

https://github.com/alextivoli/HOME_SYSTEM.git

Capitolo 1

Frontend

Per poter accedere all'applicazione un utente deve effettuare il Login ed accedere con le proprie credenziali Google. Successivamente è disponibile una pagina di controllo per gestire il sistema di temperature della stanza. È possibile monitorare la temperatura esterna e interna, accendere o spegnere la pompa di calore, aprire chiudere o aggiungere delle finestre, aprire o chiudere la porta della stanza. Inoltre è possibile monitorare l'andamento della temperatura esterna e interna nel tempo tramite due grafici. Ogni sensore ha un componente dedicato che viene creato una volta aggiunto il sensore. Il frontend è collegato al backend tramite Web Socket, da cui riceve le informazioni sui sensori e a cui inoltra i comandi. Il frontend è contenuto dentro a un container Docker.

Capitolo 2

Backend

Il backend è connesso con tutti i servizi attivi, con l'attuatore e con il frontend. Il backend è contenuto dentro a un container Docker. Raccoglie le informazioni dai sensori tramite Web Socket in cui vengono scambiati messaggi con specificato il tipo di messaggio, il mittente del messaggio e il valore contenuto che può essere stringa o un oggetto, li inoltra al frontend, e attende i comandi di un utente, i quali vengono inoltrati all'attuatore tramite chiamate Rest. I comandi possono essere: modificare lo stato della pompa di calore, della porta o di una finestra, oppure aggiungere una finestra, oppure modificare la temperatura della pompa di calore.

Capitolo 3

Microservizi

Ogni singolo microservizio all'interno dell'architettura è contenuto in un container Docker, garantendo un'implementazione modulare, scalabile e isolata delle funzionalità del sistema.

3.1 Attuatore

Il suo scopo è fare da intermediario per le chiamate REST: riceve le richieste dal backend, esegue alcuni controlli e poi inoltra la richiesta al microservizio corretto. È gestito da un microservizio che espone la porta 8084.

3.2 Pompa di Calore

Il suo scopo è mantenere aggiornato lo stato e la temperature della pompa di calore. Una pompa di calore può essere nello stato "ON", "OFF", "ERROR". Riceve comunicazione dei cambiamenti dall'attuatore, e invia informazioni al backend. È gestito da un microservizio che espone la porta 8083.

3.3 Porta

Il suo scopo è mantenere aggiornato lo stato della porta. Una porta può essere nello stato "OPEN", "CLOSED", "ERROR". Riceve comunicazione dei cambiamenti dall'attuatore, e invia informazioni al backend. È gestito da un microservizio che espone la porta 8082.

3.4 Finestre

Il suo scopo è mantenere aggiornato lo stato e la lista delle varie finestre attive. Una finestra può essere nello stato "OPEN", "CLOSED", "ERROR". Riceve comunicazione dei cambiamenti dall'attuatore, e invia informazioni al backend. È gestito da un microservizio che espone la porta 8085.

3.5 Tempo atmosferico

Il suo scopo è inviare la temperature simulata del tempo atmosferico al backend. È gestito da un microservizio che espone la porta 8081.

3.6 Termometro

Il suo scopo è mantenere aggiornata la temperatura della stanza. Riceve comunicazione dei cambiamenti di stato dei sensori e temperatura esterna o della pompa di calore tramite Web Socket dal backend e invia la temperatura aggiornata della stanza al backend. È gestito da un microservizio che espone la porta 8086.